OPEN ACCESS Freely available online

PLoS one

An Ultra-Fast Metabolite Prediction Algorithm

Zheng Rong Yang*, Murray Grant

Biosciences, College of Life and Environmental Science, University of Exeter, Exeter, United Kingdom

Abstract

Small molecules are central to all biological processes and metabolomics becoming an increasingly important discovery tool. Robust, accurate and efficient experimental approaches are critical to supporting and validating predictions from post-genomic studies. To accurately predict metabolic changes and dynamics, experimental design requires multiple biological replicates and usually multiple treatments. Mass spectra from each run are processed and metabolite features are extracted. Because of machine resolution and variation in replicates, one metabolite may have different implementations (values) of retention time and mass in different spectra. A major impediment to effectively utilizing untargeted metabolomics data is ensuring accurate spectral alignment, enabling precise recognition of features (metabolites) across spectra. Existing alignment algorithms use either a global merge strategy or a local merge strategy. The former delivers an accurate alignment, but lacks efficiency. The latter is fast, but often inaccurate. Here we document a new algorithm employing a technique known as quicksort. The results on both simulated data and real data show that this algorithm provides a dramatic increase in alignment speed and also improves alignment accuracy.

Citation: Yang ZR, Grant M (2012) An Ultra-Fast Metabolite Prediction Algorithm. PLoS ONE 7(6): e39158. doi:10.1371/journal.pone.0039158

Editor: Jérémie Bourdon, Université de Nantes, France

Received April 4, 2012; **Accepted** May 21, 2012; **Published** June 20, 2012

Funding: Data set 2 was generated from British Biotechnology and Science Research Council funding (Grants BB/C514115/1, BB/E010334/1 and BB/F005903/1). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: z.r.yang@ex.ac.uk

Introduction

Small molecules are the fundamental components of life, comprising the constituents of all biological material. Knowledge about the function, distribution and abundance of metabolites is fundamental to a comprehensive systems level understanding of an organism. Furthermore, soluble and volatile metabolites are central players in influencing interactions at a higher ecosystem level through their role in sensing, perception and elaborating biotic and abiotic stress responses. In post-genomic systems level research, the metabolome (all metabolites) of an organism is examined for various pattern analysis purposes [1] which will inform biological knowledge such as response to a particular stress or identification of molecular markers for medicinal or agricultural purposes. Multivariate analysis can be done using principal component analysis [2,3], cluster analysis [4,5], and discriminant analysis [6,7] or for differential metabolite identification [7]. As a finger-printing technique, metabolomics can support the exploration of the relationship between metabolites and interactions influencing phenotypes, driving studies on metabolite network reconstruction [8]. To ensure that these analyses are accurate and unbiased, it is necessary to make as precise a prediction of the mass and retention time of a unknown metabolite as possible. This is essential to *i*) the accuracy of compound recognition; *ii*) the accurate calculation of chemical composition of a metabolite [9]; and *iii*) the prediction of the function of unknown genes through metabolomics [8,10,11,12,13,14].

Fundamental to any biological research, dynamic behaviour of biological molecules, be they proteins, mRNA or metabolites, needs to be determined through highly replicated experimentation. Metabolite features need to be first extracted from multiple mass spectra prior to any pattern analysis. Due to machine resolution and sample variation, one metabolite will have different implementations in different spectra, i.e. non-identical retention time and mass values. This means that the exact retention time and mass values of a real, but unknown metabolites may not be seen in collected spectra. Most metabolites are unknown therefore to accurately recognize metabolites, precise alignment of features across spectra is the first critical task in analyzing metabolomic datasets based upon accurate statistical estimations.

As described recently [15], three conditions must be satisfied for aligning features. First, features must fall within defined resolutions of retention time and mass to be considered for alignment. Second, no more than two features from the same spectra can be aligned to one consensus, i.e. the collision condition (Duran, 2003). The collision problem has been long been recognised and the resolution is normally equipment-dependent [16,17,18]. Third, mass shift cannot be ignored during alignment although we commonly ignore retention time shift, which is relatively small. All are critical to a reliable prediction (alignment) for multivariate analysis [15].

There are generally two types of alignment algorithms, i.e. a local merge strategy and a global merge strategy. The former commonly employs three techniques, warping [19,20,21,22,23,24], nearest neighbour [25,26] and clustering [19,27,28]. These are generally computationally efficient, but typically scan spectra one by one to generate consensuses, which cannot be updated or revised. Consequently the first scans may generate a false consensus based on an incorrect feature set which cannot subsequently be revised when "correct" features are scanned later [15,28]. Many alignment tools, both commercial ones and freeware, belong to this type e.g. MetAlign [29], MSFACTs [26], OPenMS [30]. Binbase [31], MathDAMP [32],

ChromA [24], LC-MSsim [33], XCMS [16], SpecAlign [34], MET-IDEA [35].

In order to increase alignment accuracy we recently developed PAD (Peak Alignment via Density maximisation), which adopted a global merge strategy [15] using a concept called the Map Coverage Maximization (MCM), where a `map' refers to a spectrum. It implements a novel alignment principle, i.e. density maximisation. Among various overlapping candidate consensuses, a consensus with the highest density is selected as the prediction. A consensus refers to the prediction of a true, but unknown metabolite. However PAD is comparatively much slower than a local merge algorithm such as implemented by SIMA [28], which is typical to a global merge algorithm.

In this paper we present a novel feature alignment algorithm based upon the quicksort technique [36] used in computer sciences. The alignment run comprises four steps. The first converts features to a string list, which is then sorted. The second, similar to PAD, constructs candidate consensuses and detects their density. The third examines and filters the candidate consensuses to generate predictions. In the fourth step, features which fail to be aligned are put back to the string list and rerun. Here we evaluate this algorithm using both simulated data and real data. We conclude that this new algorithm is superior to currently available feature alignment algorithms in both alignment speed and alignment accuracy.

Results

Simulated Data – Toy A

Description of toy A is given in **Methods**. Table 1 shows the comparison of sensitivity (see **Methods** for the definition) analysis for Toy A data at noise levels 60%, 80% and 100%. For simulations with the noise levels below 60%, the sensitivity of all three algorithms is 100%. No data for specificity (see **Methods** for the definition) analysis is shown here because the specificity of all algorithms is 100%. From Table 1, we can see that at increasing noise levels (even within the allowed resolution), the sensitivity of SIMA consistently drops, from 97% to 72%, while both PAD and PASS maintains sensitivity at 100%.

Simulated Data – Toy B

Description of toy B is seen in **Methods**. No error (MH and FP - see **Methods** for the definitions) was observed for PAD and PASS for all six data sets. By contrast, when the noise level was increased from 0% to 100%, the prediction error in SIMA with the mass resolution (see **Methods** for the definition) 0.0071 Daltons got larger (Figure 1), leading to significantly increased singletons – see the trend of the first bars in Figure 1. Figure S1 shows the prediction error of SIMA with the mass resolution 0.00001 Daltons where we can see that the error is much more amplified.

Real Data

A description of the real data is given in **Methods**. Table 2 shows the comparison of the CPU performance of the three algorithms using real data. CPU was measured in seconds. The first column indicates the alignments, for instance "Col.sid.6" means aligning features of six maps for Col-0 and *sid2* at 6 hpi. The second column indicates the number of maps used for each alignment. The third column indicates the number of raw features in each alignment. The remaining three columns represent the CPU time in seconds for the three algorithms to complete the different alignments. The final column indicates the number of features reported in SIMA (mass resolution 0.0071 Daltons) outputs. The mass resolution used for running SIMA was 0.0071 Daltons. It can be seen that PASS is much faster than PAD (32 times faster) and also faster than SIMA (four times faster). It is important to note that features in original spectra files should not be duplicated nor omitted. PAD and PASS have generated alignments without these errors, however, SIMA generated alignments with duplicated and missing features. The last column of Table 2 contains the number of features reported in the SIMA output files. In theory, these numbers should concord with the numbers in column 3 of Table 2. However, 30% of raw features were missing when aligning the spectra of Col-0 and *sid2* at 6 hpi (hours post inoculation). Six duplicated features were found when aligning the spectra of Col-0 and *sid2* at 10 hpi. Six duplicated features were found when aligning the spectra of Col-0 and *sid2* at 16 hpi. Overall, 27% of features were missing when aligning 12 spectra of Col-0 at all three time points, 43% of features were missing when aligning 12 spectra of *sid2* at all three time points and the alignment of all 24 spectra delivered 17% duplicated features.

In addition, many SIMA consensuses violated the collision condition, i.e. many Type-I errors were found in SIMA alignments, e.g. containing more than one feature from the same map (spectra). Figure 2 shows the distribution of the number of duplicated maps in one consensus when aligning all 24 maps. It can be seen that the largest duplicated map number was 12, representing half of the total number of maps. Overall ~10% of consensuses predicted by SIMA (mass resolution 0.0071 Daltons) contained duplications as denoted by the first bar in Figure 2. When using a mass resolution of 0.00001 Daltons for running SIMA, no such error was observed, but other types of error were amplified - see the discussion below.

As illustrated in Figure 3, the CAM (see **Methods** for the definition) curves of PASS are always the lowest and the CAM curves of SIMA (mass resolution 0.0071 Daltons) are always the

Table 1. Sensitivity analysis of three sets (noise levels 60%, 80% and 100%) for Toy A.

60%			80%			100%		
SIMA	**PAD**	**PASS**	**SIMA**	**PAD**	**PASS**	**SIMA**	**PAD**	**PASS**
97.20%	100%	100%	86.14%	100%	100%	72.89%	100%	100%
96.17%	100%	100%	84.75%	100%	100%	73.61%	100%	100%
96.95%	100%	100%	86.70%	100%	100%	76.93%	100%	100%
96.61%	100%	100%	86.95%	100%	100%	73.98%	100%	100%
97.25%	100%	100%	84.75%	100%	100%	72.11%	100%	100%

doi:10.1371/journal.pone.0039158.t001

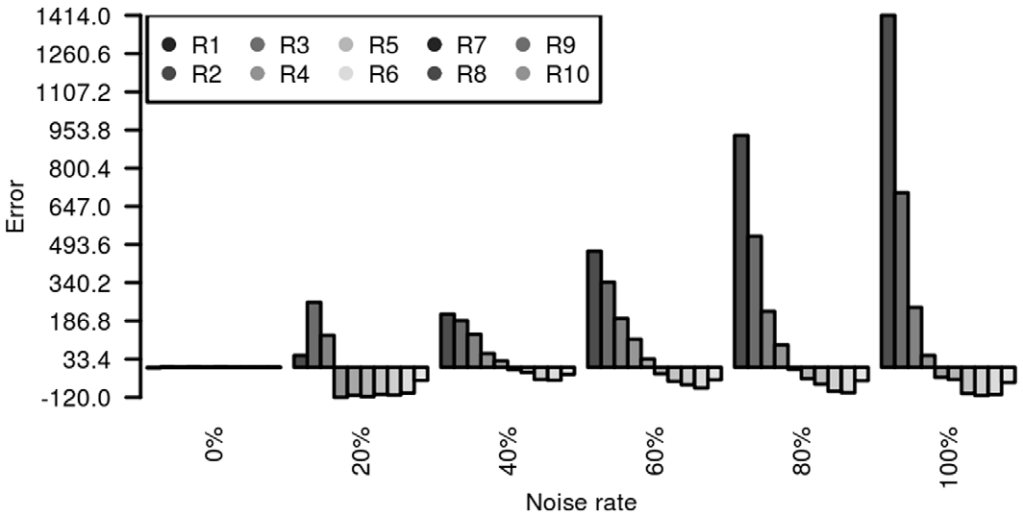

Figure 1. The distribution of prediction errors for Toy B data using SIMA (mass resolution 0.0071 Daltons). The horizontal axis represents the noise rate added to features in Toy B. The vertical axis represents either missing hypothesis (MH) or false prediction (FP). Each histogram group comprises ten bars representing ten types of consensuses (ten different number of features). The first bar represents the error between the number of expected singletons and the number of predicted singletons. The last bar represents the error between the number of true consensuses of size ten and the number of predicted consensuses of size ten. When FP occurs, we see a positive bar. When MH occurs, we observe a negative value.
doi:10.1371/journal.pone.0039158.g001

highest. Notably four plots of SIMA show flat sections at the top, meaning that for these alignments, no large consensuses were generated, which was defined as the pattern IV (the biased H-pattern) in **Methods**. Figure S2 shows a comparison when running SIMA based on the mass resolution 0.00001 Daltons, where we can see that all CAM curves of SIMA are similar to the poorest performance, which was defined as the pattern I (the disastrous pattern) in **Methods**.

The objective of improving alignment quality is to improve the quality of subsequent multivariate analysis. Accompanying this new alignment algorithm, we also introduce a novel significance analysis. Three widely used significance analysis algorithms; SAM [37], eBayes [38], and Cyber-T [39] were employed. The R program for detecting significantly differential metabolites is included in http://ecsb.ex.ac.uk/PASS. The prediction of significantly differential metabolites (between the *Arabidopsis* Col-0 wild type plant and salicylic acid deficient *sid2* mutant in this paper) was done via the consensus among the three algorithms. Figure 4 shows the distribution of significantly differential metabolites at 6 hpi, 10 hpi and 16 hpi. The use of this consensus approach can minimize the chance of a false prediction of differential metabolites because the three tests often disagree in terms of tail probabilities – small p values. Figure S3 illustrates such an example. With a simple consensus approach, we select predictions agreed by all three algorithms under a given significance level. In this study the significance level was set at 0.001 (this can be varied by the user when using our R code) leading to 11, 14 and 2 significantly differential metabolites for these three aligned data. They were shown as vertical lines in Figure 4. It should be noted that a metabolite with the largest mean differential abundance is not necessarily guaranteed to be predicted as being significantly differential. This is because the prediction does not only rely on the mean differential abundance, but also the variance. Here a differential abundance is the difference between the abundances of two treatments for a metabolite.

The accompanying R program also supports locating significantly differential metabolites in a R-M (Retention time – Mass) density surface, i.e. where we can visualize the relationship between detected significantly differential metabolites and retention-time mass density. Figure 5 shows three plots for this visualization function.

Figure S4 illustrates the usage of the PASS program.

Table 2. Comparison of CPU times for the three algorithms to generate six alignments of the real data representing metabolite changes in *Arabidopsis thaliana* leaves infected with virulent *Pseudomonas syringae*.

Alignment	# maps	# features	SIMA CPU	PAD CPU	PASS CPU	SIMA prediction
Col.sid.6	6	54330	51	81	8	37767
Col.sid.10	6	56117	56	87	9	56123
Col.sid.16	6	66285	76	124	18	66291
Col	12	109369	244	1150	53	79301
Sid	12	119866	262	1544	57	68865
All	24	229235	1541	11598	360	267811

doi:10.1371/journal.pone.0039158.t002

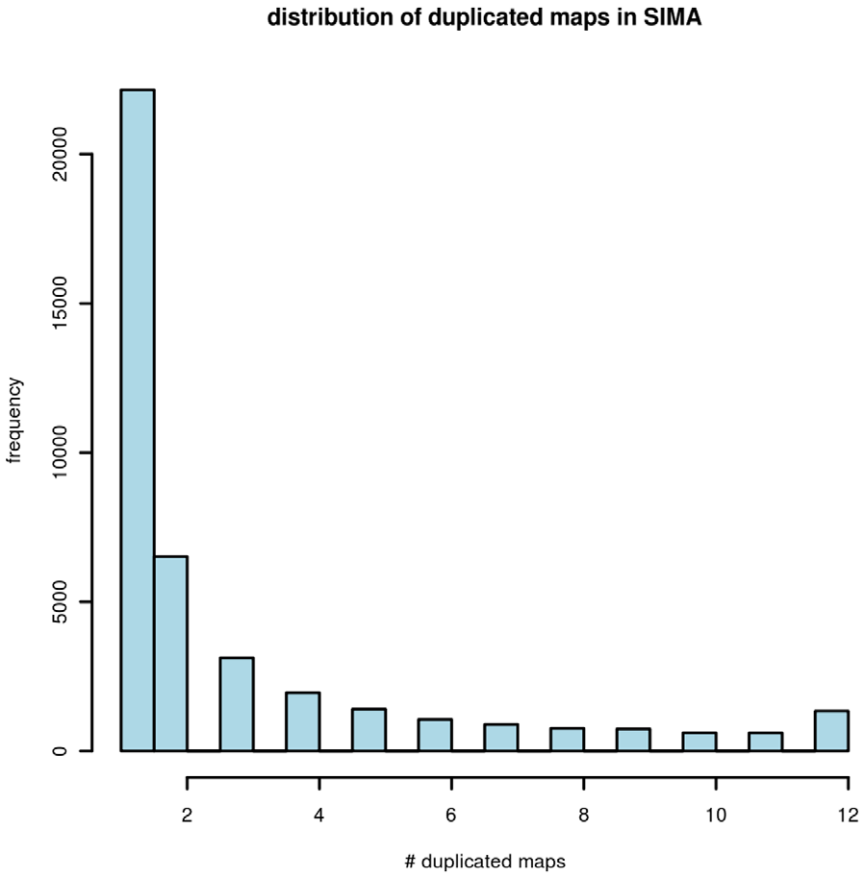

Figure 2. The distribution of duplicated maps in SIMA (mass resolution 0.0071 Daltons) consensuses alignment based on all 24 maps. The horizontal axis represents the number of Type-I errors in the generated consensuses. These range from one to 12. The vertical axis represents the frequency.
doi:10.1371/journal.pone.0039158.g002

Discussion

This paper has presented a new metabolite prediction (mass feature alignment) algorithm based on a widely used concept in computer sciences, the quicksort technique. The objective was to maintain the alignment accuracy based on the map coverage maximization principle, as recently described by Perera *et al.* in PAD (Perera et al. 2011), and to speed up alignment. PAD adopts a global merge strategy in contrast to many local merge algorithms, giving an improved alignment accuracy. Because a local merge algorithm has no regression process, its alignment is often problematic leading to poor alignment quality, which has two consequences, i.e. duplication and unreliable alignment. This was demonstrated here using SIMA, a typical local merge strategy algorithm. While a local merge algorithm is computationally fast, PAD, a typical global merge algorithm is not. We therefore implemented a quicksort approach, which is used in many programming languages, to speed up the global merge algorithm. Here we have built alternately M-clusters and R-clusters based on sorted mass and retention time values. Prior to building these two types of clusters, we converted all the numerical data including mass, retention time, metabolite abundance and spectra index to strings and organized them into a string list with recognizable labels to discriminate them. Applying the quicksort technique based on mass or retention time will not affect other domains of data and maintains a feature's spectra index and abundance value during sorting. We additionally proposed a new technique for quantifying the quality of an alignment, i.e. Characteristic Alignment Map (CAM). Using CAM analysis, the alignment quality can be easily visualized qualitatively between different alignments. We have compared this new algorithm against PAD and SIMA using toy data sets and demonstrated that this new algorithm has improved alignment accuracy. Furthermore, we have shown using a real dataset that this algorithm has significantly improved alignment quality compared with SIMA and also has a better performance than PAD. Importantly, this new algorithm is 32 times faster than PAD and SIMA. The speed improvement has also been demonstrated theoretically in **Remark 3**. The most important concept for a global optimization process for peak alignment is consensus generation. Based on this study and our earlier work on PAD, it can be seen that a consensus must be a cluster of peaks with similar mass values and retention times which satisfy the resolution condition as well as the collision condition. Local optimization, as we have shown, will not be able to find all these peaks for one consensus. However comparing all peaks one by one is a typical NP (non-deterministic polynomial-time) - hard problem [43] as we saw in PAD. This is why the quicksort technique can significantly reduce the complexity leading to successful global optimization. Accompanying this alignment algorithm, we also introduced a novel approach for detecting significantly differential metabolites using a simple consensus principle to minimize the chance of delivering falsely predicted differential metabolites and visualizing the detected significantly differential metabolites.

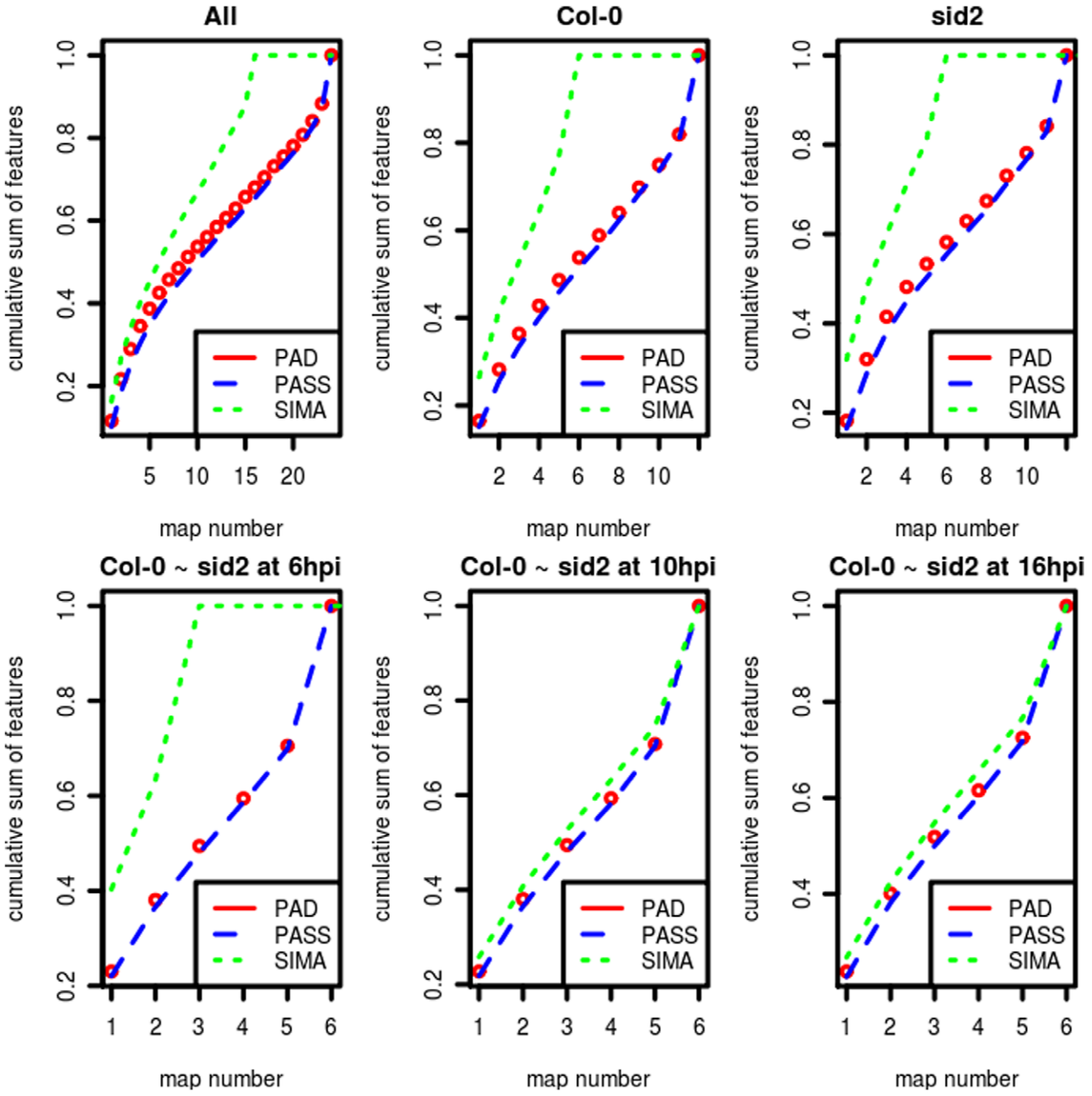

Figure 3. Characteristic alignment map (CAM) curves. The CAM was generated for MCM analysis of six alignments on the real data of pathogen infected plant leaves. The horizontal axes represent the maps used for each alignment, i.e. from six to 24. The vertical axes represent the cumulative sum of aligned features or the size of consensuses. The open dots represent CAM curves of PAD. Dashed lines represent CAM curves of PASS and dotted lines represent CAM curves of SIMA (mass resolution 0.0071).
doi:10.1371/journal.pone.0039158.g003

Methods

Algorithm

The notations used by the algorithm are as follows: A data set is denoted by $\mathbf{X}$, which is composed of N discrete features of K maps. Each map refers to a mass spectrum. Each feature $\mathbf{x}_i \in \mathbf{X}$ is a vector of four values, i.e. retention time r_i, mass m_i, map index w_i and feature intensity (abundance) z_i. Retention time and mass reflect the chemical property of a metabolite and are used for predicting the chemical composition of a compound. The feature intensity is the reflection of the abundance of a metabolite and is the main parameter used in multivariate analysis, most notably differential metabolite predictions. The map index is only used to classify features, i.e. indicating from which spectrum a feature is collected. In addition to feature intensity, both r_i and m_i contain variation arising from both experimental and mass spectral resolution variation. The extent of variation is usually known.

It is also assumed that the observed features are random samples of a true, but unknown metabolite. This means that the following condition should be satisfied for an alignment of each feature

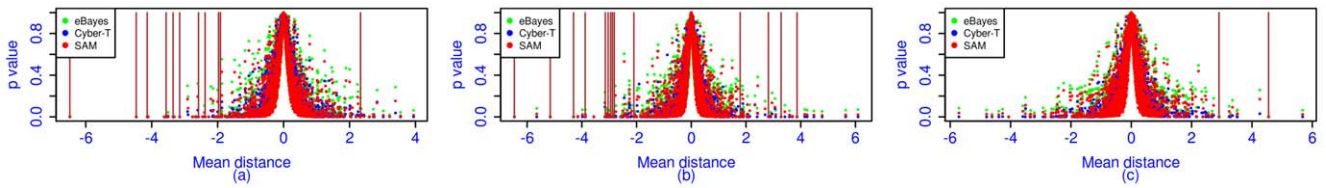

Figure 4. Significantly differential metabolites identified between Col-0 and *sid2* leaves responding to infection with *P. syringae* at 6 hpi, 10 hpi and 16 hpi. The horizontal axes represent the mean distance between Col-0 abundance and *sid2* abundance. The vertical axes represent *p* values. Each dot represents one metabolite. Each vertical line represents a significantly differential metabolite. (a) - top: Significantly differential metabolites between Col-0 and *sid2* at 6 hpi. (b) - middle: Significantly differential metabolites between Col-0 and *sid2* at 10 hpi. (c) - bottom: Significantly differential metabolites between Col-0 and *sid2* at 16 hpi.
doi:10.1371/journal.pone.0039158.g004

$$|\tilde{\mathbf{x}}_i - \mathbf{u}_k| \le \varepsilon_k \quad (1)$$

where $\tilde{\mathbf{x}}_i = (r_i, m_i)$ is the retention time - mass pair of a feature, which is an observed metabolite in a spectrum, $\mathbf{u}_k = (\bar{r}_k, \bar{m}_k)$ is the retention time - mass pair of a true metabolite, and $\varepsilon_k = (\varepsilon_r, \varepsilon_m^k)$ is the pre-defined resolution set (retention resolution and mass resolution). Here ε_r is commonly a constant (0.3 in this paper according to our mass spectrometer resolution) and ε_m^k is variable, i.e. $\varepsilon_m^k = \varepsilon_m^0 \times \bar{m}_k$. ε_m^0 is a constant (10 ppm (its corresponding mass resolution is 0.00001 Daltons) in this paper as constrained by our mass spectrometer resolution), and $\bar{m}_k$ is the k^{th} true mass under estimation. As each map may contain tens of thousands features, aligning features from many spectra becomes problematic in terms of speed - see Table 2.

Here we adopt a different strategy to speed up an alignment process dramatically while maintaining the alignment accuracy. In this algorithm, we still follow the resolution condition described in equation (1) and the collision condition. Following [15], we assume that the mass shift is linearly proportional to the true mass, i.e.

$$|m_i - \bar{m}_k| \le \varepsilon_m = \varepsilon_m^0 \times \bar{m}_k \quad (2)$$

In theory, $\bar{r}_k$ and $\bar{m}_k$ may not be exactly estimated. We therefore use their estimations, i.e. $\hat{r}_k$ and $\hat{m}_k$, in an alignment process. A consensus is then expressed by $(\hat{r}_k, \hat{m}_k)$.

The quicksort technique, a well known algorithm in computer sciences and implemented as a basic function in various programming languages, such as C, is used here to implement our algorithm. It sorts strings in a lexicographical order, i.e. the difference at an earlier position of strings has a priority compared with differences occurring at a latter position of strings. For instance, three strings AATT, ABAA and AAAA will be sorted to an order such as AAAA, AATT and ABAA. If strings represent numerical data, the order reflects the numeric accuracy of similarity, e.g. 130.034, 130.411, 130.410, 130.029, 130.411, 130.409, and 130.035 leads to 130.029, 130.034, 130.035, 130.409, 130.410, 130.411, and 130.411. At a mass resolution of 0.001, we can easily identify two clusters; (i) 130.033, 130.034, and 130.035 with the centre as 130.034 and (ii) 130.409, 130.411 and 130.410 with the centre as 130.410. The algorithm presented here was motivated by this observation. We note that this has been previously applied to proteomics studies [40,41], where a single peptide mass was used for a targeted search within a data set of masses.

Mass spectral feature alignment is conducted in a two-dimensional space, reporting retention time and mass. We first designed a novel data structure to convert $\mathbf{X}$ to a string list $\mathbf{S}$ in which each feature is expressed using a string

$$\mathbf{s}_i = m_i \$ r_i \$ w_i \$ z_i \quad (3)$$

where $\mathbf{s}_i \in \mathbf{S}$. Using this notation, the dollar mark is used to separate four data domains. The use of the dollar mark will not affect a

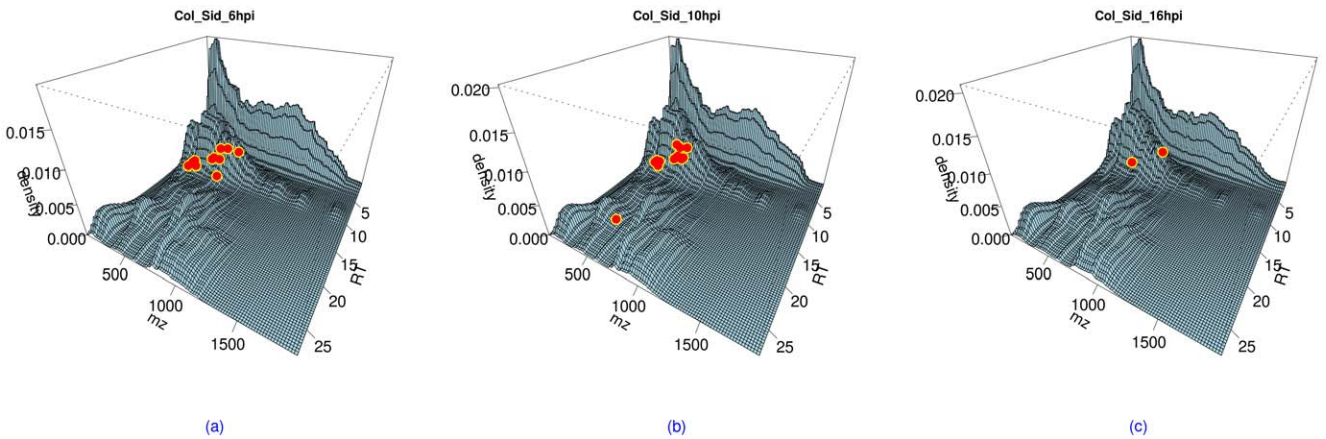

Figure 5. The location of significantly differential metabolites in R-M density surface. The significantly differential metabolites were shown using dots on the surfaces. (a) - left: for significantly differential metabolites between Col-0 and *sid2* at 6 hpi. (b) - middle: for significantly differential metabolites between Col-0 and *sid2* at 10 hpi. (c) - right: for significantly differential metabolites between Col-0 and *sid2* at 16 hpi.
doi:10.1371/journal.pone.0039158.g005

sorting process based on mass, which is at the first domain in the string list.

In order to guarantee an accurate sorting of data, all numerical data must be of the same length. If a feature's retention time (or mass) has lower than the maximal number of digits (decimals) then '0' is introduced to enable the sorting to function appropriately (e.g. 1.5 becomes 001.5000 if the maximum number of digits is three and the maximum number of decimals is four). We refer to such a numerical value (say 001.5000) as a digit-aligned-value (DAV).

The alignment is run in two stages. In the first stage, we construct so-called mass clusters or M-clusters. Each M-cluster is composed of a number of features, which satisfy the enlarged mass resolution,

$$|m_i - m_j| \le 2 \times \varepsilon_m^i \tag{4}$$

where $\varepsilon_m^i = \varepsilon_m^0 \times m_i$. Figure 6 illustrates how a mass cluster is constructed, where retention time, map index and feature intensity are masked, hence not being used for the construction of this M-cluster.

An M-cluster is constructed by sequentially scanning the string list **S** till equation (4) is violated. For Figure 6, the scan was terminated or the M-cluster is constructed between i^{th} feature and the j^{th} feature if

$$|m_i - m_{j+1}| > 2 \times \varepsilon_m^i \tag{5}$$

The resolution is doubled in equation (5) because m_i and m_j can be just on the two extreme boundaries of a consensus, i.e.

$$|m_i - \hat{m}_k| = \varepsilon_m^k \;\&\; |m_j - \hat{m}_k| = \varepsilon_m^k \tag{6}$$

where $\hat{m}_k$ is the median mass of the kth consensus. Remark 1 below shows that this strategy is safe to construct an M-cluster as well as an R-cluster later. In addition, together with equation (7) given below, we call this strategy greedy scanning. Remark 2 below shows that this strategy almost guarantees the formation of an unbiased consensus. Staring from the j +1th string in S, the next M-cluster can be constructed. For each M-cluster, which is denoted by $\theta = (\mathbf{s}_i, \cdots, \mathbf{s}_j) \subset \mathbf{S}$, the second stage of this algorithm is to examine the retention time of the strings in θ to construct retention time clusters or R-clusters. Note that there might be a number of R-clusters in one M-cluster because different consensuses may share very similar retention times as discussed in [15]. Prior to constructing R-clusters within one M-cluster, we have to move into another string structure to enable sorting retention time. In order to avoid any incorrect manipulation of the string list, we have to target this M-cluster locally. In practice, we simply copy the M-cluster to another string list shown in Figure 7, where we insert one more column ("o") to remember where each feature (string) is copied from the S list. This reduced list is called a θ-list.

After sorting the retention time in the θ-list, the original order of strings in the θ-list will be changed. The use of the "o" column in this reordered θ-list (Figure 8) will save the information of the indexes to the **S** list, which is critical for later manipulations. As all the data including mass, map index, and feature intensity of a string (feature) are unchanged, these will shift concomitantly as string positions are resorted.

We next focus on forming R-clusters in the sorted θ-list. Starting from the first string in a sorted θ-list, we scan features one by one to examine if the condition described below is satisfied

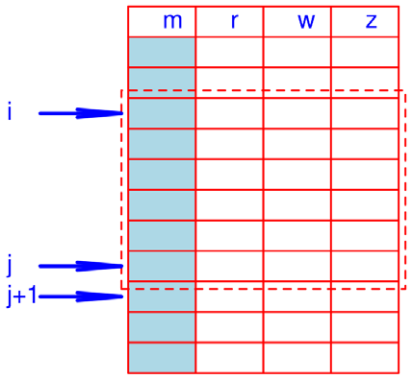

Figure 6. An illustration of constructing a M-cluster. The title line indicates the four fields of the string list; "*m*" stands for mass, "*r*" stands for retention time, "*w*" stands for map index, and "*z*" stands for feature intensity. The M-cluster starts from the i^{th} string (row) and ends at the j^{th} string (row). The dashed box indicates that the features (strings) within it form the M-cluster.
doi:10.1371/journal.pone.0039158.g006

$$|r_i - r_{j+1}| > 2 \times \varepsilon_r \tag{7}$$

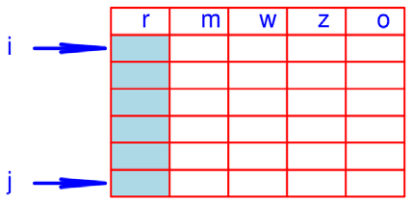

Figure 7. R-cluster formation in an θ-list. The first column stores retention time values of all features in the θ-list. In addition to four columns, we have introduced the "o" column for indexing the **S** list.
doi:10.1371/journal.pone.0039158.g007

We similarly double the retention time resolution as above because r_i and r_j can reside on the two extreme boundaries of a consensus, i.e.

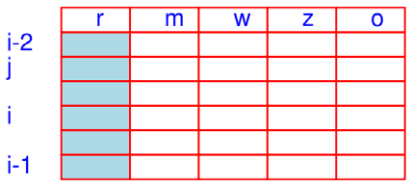

Figure 8. The θ-list after sorting based on retention time.
doi:10.1371/journal.pone.0039158.g008

$$|r_i - \hat{r}_k| = \varepsilon_r \;\&\; |r_j - \hat{r}_k| = \varepsilon_r \tag{8}$$

where $\hat{r}_k$ is the median retention time of the kth consensus. Staring from the j +1th string in a sorted θ-list, a next R-cluster will be considered. For each R-cluster denoted by $\pi = (\mathbf{s}_i, \cdots, \mathbf{s}_j) \subset \theta$, a consensus is constructed. For all features in π, we calculate its median mass and median retention time using the following definition

$$\hat{r}_k = \frac{1}{2}(r_i^- + r_i^+),\ \forall \mathbf{s}_i \in \pi\ \&$$
$$\hat{m}_k = \frac{1}{2}(m_i^- + m_i^+),\ \forall \mathbf{s}_i \in \pi \quad (9)$$

where r_i^- and r_i^+ are the minimum and maximum retention times among all features in the current R-cluster (π). m_i^- and m_i^+ are the minimum and maximum masses among all features in π. Deriving median mass and median retention time this way is designed to avoid possible bias [15].

To save computing time, we always remove all the aligned features from the **S** list every time prior to running quicksort. To do so, we simply "whiten" all the strings corresponding to the aligned features by replacing the mass by the letter "w". As the "o" column in the θ-list records the original positive in the **S** list, it is very easy to trace them back to the **S** list to whiten the corresponding strings. After using the quicksort technique, all the strings of the aligned features (hence whitened ones) will be moved to the bottom of the **S** list automatically and will not be visited in subsequent scans (Figure 9).

When constructing a consensus, we need to mitigate two types of errors. A type-I error occurs when two features satisfy the resolution defined in equation (1) but are in the same spectra (map). A type-II error refers to the situation when a feature in a cluster does not satisfy the resolution defined in equation (1).

In order to follow the Map Coverage Maximization (MCM) principle [15], we first construct consensuses which cover all maps. When no further consensus can be constructed, we then look for consensuses, which cover n - 1 maps. This is repeated till one map is left. For instance, we will start finding consensuses of size ten if the total number of spectra is ten. If no consensus of size ten can be found, we search for consensuses of size nine, etc. In this way, we can ensure that the MCM principle is followed to generate reliable alignments.

The algorithm is implemented in C based on a linux computer with 3GB memory of 2.6 Ghz. The executable code is available at http://ecsb.ex.ac.uk/PASS.

m	r	w	z
w			
w			
w			

Figure 9. Example of "whitening" strings corresponding to aligned features. The rows with the "w" letter represent the strings of aligned features. Following quicksort these rows will be at the bottom of the **S** list and will not be re-visited in subsequent scans.
doi:10.1371/journal.pone.0039158.g009

Remark 1

ℓ DAVs in a sorted list corresponding to ℓ numerical values $\{z_i\}_{i=1}^{\ell}$ always follow a sequence of $z_{(1)} \le z_{(2)} \le \cdots \le z_{(\ell)}$, where $z_{(i)}$ is the i^{th} DAV in the sorted list.

Proof: We use the *reductio ad absurdum* approach for this proof. Suppose $z_{(i)} > z_{(j)}$, but $z_{(i)} \lhd z_{(j)}$. Here we use $\lhd$ to denote an ascending order or lexicographical order, i.e. $z_{(i)}$ precedes to $z_{(j)}$ in a DAV list. For simplicity, we assume all values in a DAV list are integers. Generalizing the proof for values with decimals is straightforward. Suppose $k \in [1, D]$ with D as the length of all DAVs is the first digit makes $z_{(i)}$ and $z_{(j)}$ different. For instance, if two DAVs are 01312 and 01322, $k = 3$ and $D = 4$. We denote the two letters of these two DAVs at this position as $z_{(i),k}$ and $z_{(j),k}$. If $z_{(i)} > z_{(j)}$, it is almost certain that $z_{(i),k} \rhd z_{(j),k}$. This means that $z_{(i)} \lhd z_{(j)}$ is not possible.

Remark 2

The greedy scanning guarantees the formation of a consensus of all its features for a sorted list of mass and retention time values.

Proof: Again, we use the *reductio ad absurdum* approach for this proof. Suppose a feature list $\sigma = (z_1, z_2, \cdots, z_\ell)$ forms a consensus ($\ell \le L \le K$ - K is the number of maps) and a sorted DAV list of it is expressed as $\tilde{\sigma} = (z_{(1)}, z_{(2)}, \cdots, z_{(\ell)})$, where $z_{(i)}$ is the i^{th} DAV in the sorted list. Based on the assumption that σ forms a consensus, $|z_{(1)} - z_{(\ell)}| \le 2\varepsilon$ and $|z_i - z_j| \le 2 \times \varepsilon$, $\forall z_i, z_j \in \sigma$, where $\varepsilon = \varepsilon_m$ or $\varepsilon = \varepsilon_r$. If one feature (denoted by $z_s \in \sigma$) is beyond the cluster, it means that $z_s \lhd z_{(1)}$ or $z_s \rhd z_{(\ell)}$. In other words, $|z_s - \min(\sigma)| > 2 \times \varepsilon$ or $|z_s - \max(\sigma)| > 2 \times \varepsilon$. This is contrary to the assumption.

Remark 3

The average time complexity of PASS follows $O(\text{PASS}) \propto N \log N$ [42].

Proof: The time complexity of quicksort is $N \log N$. As it is difficult to estimate the metabolite distribution, we first assume that the features are equally distributed for consensuses of different size, i.e. the features are equally divided to form consensuses covering different numbers of maps. Importantly; *i*) we whiten corresponding strings in the **S** list whenever a consensus is formed; *ii*) quicksort is only used when the S list is exhausted. This means that the number of the strings in the **S** list when calling quicksort is decreased step by step as shown below (a note to the following equation is seen "**A NOTE TO REMARK 3**" in the supplementary document)

$$N \log N + N \frac{K-1}{K} \log N \frac{K-1}{K} + \cdots + N \frac{2}{K} \log N \frac{2}{K} + \frac{N}{K} \sum_{k=2}^{K} k$$

or

$$N \log N + \frac{N}{K} \sum_{i=2}^{K-1} i \log N \frac{i}{K} + \frac{N}{K} \sum_{k=2}^{K} k$$

where the second component can be further re-written as

$$\frac{N}{K} \sum_{i=2}^{K-1} i[\log N - \log(i/K)] = \frac{N}{K} \sum_{i=2}^{K-1} i[\log N - \log(i/K)]$$
$$< N \log N - \sum_{i=2}^{K-1} i \log(i/K)$$
$$< N \log N$$

the last component of the above equation can be simplified as $NK < N \log N$ - see **REMARK S1** in the supplementary document. We next assume that all features contribute to singletons. In this case, only one quicksort is required and one scanning process of the **S** list is required. It is not difficult to see that the time complexity is $N \log N + N$. We finally assume that all features contribute to consensuses with full size, i.e. covering all maps. Following the **REMARK** 2 discussed above, it can be seen that only one call to quicksort can guarantee the formation of all consensuses.

Simulated Data Preparation

In addition to the simulated data used in PAD [15] (Toy B), an additional data set (Toy A) comprising two maps was used in this paper. In this new data set, "true simulated metabolites" (TSMs) were randomly generated using a retention time between 1 min and 27 min as well as mass between 1 and 500 following [15]. Two categories of TSMs were designed, i.e. non-aligned or aligned. Only two maps (spectra) were generated for analyzing both prediction sensitivity and specificity. The sensitivity is the percentage of aligned TSMs that are correctly aligned. The specificity is the percentage of non-aligned TSMs that are not aligned. For a non-aligned TSM, a feature was generated through adding random noise to both retention time and mass. These noise levels were sequentially 20%, 40%, 60%, 80% and 100% of the given resolution [15]. A feature of a non-aligned TSM was generated by

$$m_i = \bar{m}_i + U(\lambda \times \varepsilon_m^0 \times \bar{m}_i) \tag{10}$$

and

$$r_i = \bar{r}_i + U(\lambda \times \varepsilon_r \times \bar{r}_i) \tag{11}$$

where $U(\lambda \times \varepsilon_r \times \bar{r}_i)$ is the uniform distribution function with the interval defined as $[-\lambda \times \varepsilon_r \times \bar{r}_i, \lambda \times \varepsilon_r \times \bar{r}_i]$, and $\lambda \in \{0.2, 0.4, 0.6, 0.8, 1\}$ represents the noise level. The feature generated this way was then randomly distributed into one of two maps. For an aligned TSM, which in this case contains two features (because Toy A has two maps), each feature was generated through adding random noise and was distributed into one of two maps. Each feature was formed by both mass

$$m_{ik} = \bar{m}_i + U(\lambda \times \varepsilon_m^0 \times \bar{m}_i) \tag{12}$$

and retention time

$$r_{ik} = \bar{r}_i + U(\lambda \times \varepsilon_r \times \bar{r}_i) \tag{13}$$

where λ is as described above and $k \in \{1,2\}$. Figure 10 shows the distributions of features of one such data set, where 493 aligned TSMs (comprising 986 features) and 504 non-aligned TSMs were generated.

Real Data Preparation

The data from [15] was used in this study for the comparison. The data is seen in ecsb.ex.ac.uk/PASS.

Comparison of Algorithms

We used SIMA [28] and PAD [15] to evaluate the new algorithm as they represent the current benchmark for this type of application. Following [15], two mass resolutions (0.0071 Daltons and 0.00001 Daltons) were used to run SIMA for comparison one mass resolution (0.00001 Daltons) was used to run PAD and PASS. SIMA does not consider mass shift. We therefore follow PAD to use two mass resolutions for comparison.

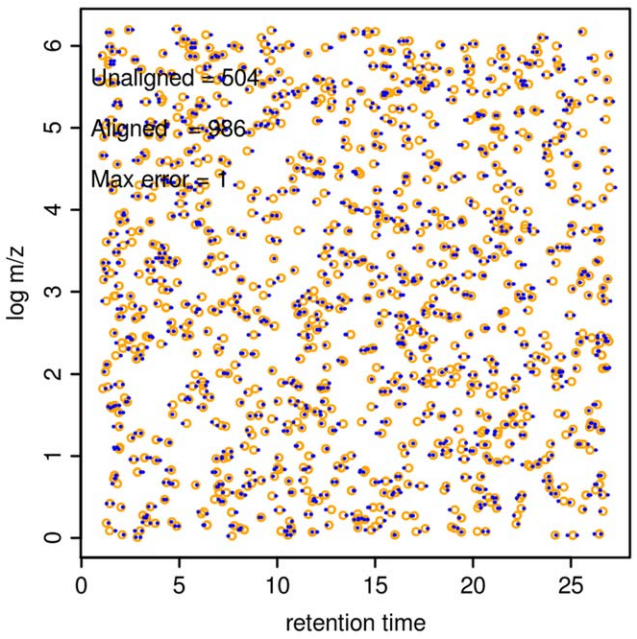

Figure 10. Distributions of features and TSMs in Toy A data. The circles represent TSMs and the dots represent the features in two maps. The two axes represent retention time and log*m/z* (or mass). The three lines of texts in the plots represent, in order; *a*) the number of features (non-aligned TSMs), which should not be aligned; *b*) the number of features (aligned TSMs), which should be aligned; *c*) maximum allowed noise level. A value of "1" means that noise was added to features at the maximum 100% of the pre-defined resolution, i.e. 0.3 min for retention time and 10 ppm for mass.
doi:10.1371/journal.pone.0039158.g010

Sensitivity/specificity Analysis

To compare algorithms for these criteria we limited our analysis to Toy A data. We used the following assumptions. Suppose the number of non-aligned features is N and number of aligned features is $2P$, P being the number of TSMs. If the observed number of singletons is N_0 and the number of aligned consensuses is C_0, then specificity is defined as

$$\mathrm{SPE} = 100\frac{N_0}{N}\% \tag{14}$$

and the sensitivity is defined as

$$\mathrm{SEN} = 100\frac{C_0}{P}\% \tag{15}$$

Prediction Error – Missing Hypothesis (MH) and False Prediction (FP)

An alignment may introduce two prediction errors; a missing hypothesis (MH) or a false prediction (FP). A missing hypothesis means that a consensus of a specific size is lost during alignment (prediction). A false prediction means that an incorrect consensus is introduced for a specific consensus size. For simulated data (Toy B), we know in advance how many consensuses are expected. Post alignment, we have a set of consensuses, each formed by different

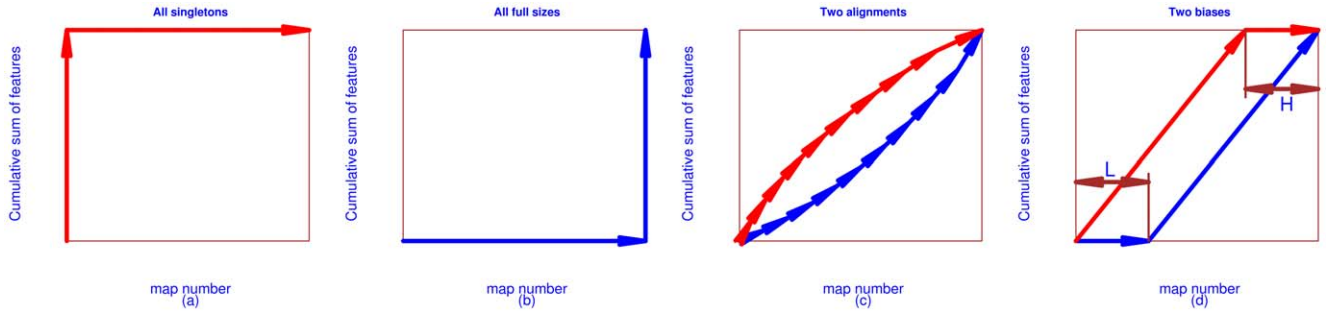

Figure 11. Two extreme and common examples of CAM curves. (a) Pattern I (disastrous pattern): all predicted consensuses are singletons; (b) Pattern II (perfect pattern): all predicted consensuses are of full size; (c) Pattern III (normal pattern): the comparison of two CAM curves for two alignments; (d) Pattern IV (biased pattern): two biased alignments. The upper one is defined as the biased H-pattern and the lower one is defined as the biased L-pattern. The horizontal axes represent the number of maps. The vertical axes represent the cumulative sum of features. (a) - panel 1: Pattern I; (b) - panel 2: Pattern II; (c) - panel 3: Pattern III; (d) - panel 4: Pattern IV.
doi:10.1371/journal.pone.0039158.g011

numbers of features, corresponding to the consensus size. Suppose we have K maps, we use the following notation to denote the number of consensuses from 1 to K sizes, $\mathbf{c}=(c_1,c_2,\cdots,c_i,\cdots,c_K)$, where c_i represents the number of consensuses of size i. In addition to the $\mathbf{c}$ vector, we define another vector of TSMs, $\mathbf{t}=(t_1,t_2,\cdots,t_i,\cdots,t_K)$, where t_i represents the number of TSM of size i. MHs occur when

$$t_i > c_i \tag{16}$$

and FPs occur when

$$t_i < c_i \tag{17}$$

Note that this measure only applies to a simulated data set where the t vector is known.

Characteristic Alignment Map (CAM)

We introduced this for comparing algorithms on real data. Based on the $\mathbf{c}$ vector, we calculated the cumulative sum of features aligned to different consensus sizes. It was denoted by $\alpha=(\alpha_1,\alpha_2,\cdots\alpha_i,\cdots\alpha_K)$ and α_i was defined by

$$\alpha_i=\sum_{j=1}^{i} j\times c_j \tag{18}$$

We used the map number as the horizontal axis and α as the vertical axis to plot the data of α. We referred to α as the characteristic set and referred to this plot as a Characteristic Alignment Map (CAM) curve. In the worst case scenario, all predicted consensuses are singletons, i.e. being composed of a straight line in concord with the vertical axis first and a straight line in concord with the horizontal axis next – Figure 11 (a). This pattern is defined as Pattern I - disastrous pattern. A perfect alignment should generate CAM a curve touching the bottom-right corner, i.e. being composed of a straight line in concord with the horizontal axis first and a straight line in concord with the vertical axis next – Figure 11 (b). This pattern is defined as Pattern II - perfect pattern. Because many consensuses don't occupy all maps, a CAM curve will stretch from the bottom-right corner towards to the top-left corner, i.e. between the two extreme curves - Figure 11 (c). This pattern is defined as Pattern III - normal pattern. In comparison, an alignment with a lower CAM curve is preferred compared with an alignment with a higher CAM curve, for instance the lower CAM curve in Figure 11 (c) is preferred. In Figure 11 (d), we show two biased alignments. They are defined as Patterns IV - biased patterns. The higher CAM curve shows the situation that the alignment losses consensuses with large sizes - H-pattern. If the map number is M, the alignment generates zero consensuses with sizes from M - H to M. The lower CAM curve illustrates that the alignment has no consensuses with small sizes - L-pattern. For map number M, the alignment generates zero consensuses with sizes from one to L. In theory, the total number of features before and after alignment should be identical. As SIMA was not reliable in this respect, the characteristic set (see **Methods** for details) was normalized for each algorithm in this paper for comparison, i.e.

$$\tilde{\alpha}_i=\frac{\alpha_i}{\sum_{j=1}^{K}\alpha_j} \tag{19}$$

where K refers to the number of maps (spectra). We then used $\tilde{\alpha}=(\tilde{\alpha}_1,\tilde{\alpha}_2,\cdots\tilde{\alpha}_i,\cdots\tilde{\alpha}_K)$ to investigate which alignment best follows the MCM rule [15].

Supporting Information

Figure S1 The distribution of prediction errors for Toy B data using SIMA (mass resolution 0.00001 Daltons). The horizontal axis represents the noise rate added to features in Toy B. The vertical axis represents either missing hypothesis (MH) or a false prediction (FP). Each histogram group comprises ten bars representing ten types of consensuses, i.e. consensuses containing ten different features. The first bar represents the error between the number of expected singletons and the number of predicted singletons. The last bar represents the error between the number of true consensuses of size ten and the number of predicted consensuses of size ten. When FP occurs, we will see a positive bar (extending upwards from the horizontal axis). When MH occurs, we observe a negative value (extending downwards from the horizontal axis).
(TIFF)

Figure S2 Characteristic alignment map (CAM) curves. The CAM was done for MCM analysis of six alignments on the real data of pathogen infected plant leaves. The horizontal axes represent the maps used for each alignment, i.e. from six to 24. The vertical axes represent the cumulative sum of aligned features or the size of consensuses. The open dots represent CAM curves of

PAD. Dashed lines represent CAM curves of PASS and dotted lines represent CAM curves of SIMA (mass resolution 0.00001 Daltons).
(TIFF)

Figure S3 *p* value distributions of three modified *t* tests. Both horizontal and vertical axes represent *p* values ranging from zero to one.
(TIFF)

Figure S4 Instructions for using PASS.
(TIF)

Remark S1
(DOC)

Acknowledgments

The authors thank Marta De Torres Zabala for the biological data, Venura Perera, Hannah Florance and Nicolas Smirnoff for helping on generating the mass data. Thanks also to Zi Hua Yang for the discussion of mathematics of this algorithm.

Author Contributions

Conceived and designed the experiments: ZRY. Performed the experiments: ZRY. Analyzed the data: ZRY MG. Contributed reagents/materials/analysis tools: ZRY MG. Wrote the paper: ZRY MG.

References

1. Biedendieck R, Borgmeier C, Bunk B, Stammen S, Scherling C, et al. (2011) Systems biology of recombinant protein production using Bacillus megaterium. Methods Enzymol 500: 165–195.
2. Ma Q, Zhou J, Zhang W, Meng X, Sun J, et al. (2011) Integrated proteomic and metabolomic analysis of an artificial microbial community for two-step production of vitamin C. PLoS One 6: e26108.
3. Westergaard S, Oliveira AP, Bro C, Olsson L, Nielsen J (2007) A systems biology approach to study glucose repression in the yeast Saccharomyces cerevisiae. Biotechnol Bioeng 96: 134–145.
4. Matthäus F, Salazar C, Ebenhöh O (2008) Biosynthetic potentials of metabolites and their hierarchical organization. PLoS Comput Biol 4: e1000049.
5. DiMaggio PJ, McAllister SR, Floudas CA, Feng XJ, Rabinowitz JD, et al. (2008) Biclustering via optimal re-ordering of data matrices in systems biology: rigorous methods and comparative studies. BMC Bioinformatics 9: 458.
6. Okada T, Afendi FM, Altaf-Ul-Amin M, Takahashi H, Nakamura K, et al. (2010) Metabolomics of medicinal plants: the importance of multivariate analysis of analytical chemistry data. Curr Comput Aided Drug Des 6: 179–196.
7. Robinson A, Ukrainetz NK, Kang KY, Mansfield SD (2007) Metabolite profiling of Douglas-fir (Pseudotsuga menziesii) field trials reveals strong environmental and weak genetic variation. New Phytol 174: 762–773.
8. Mapelli V, Olsson L, Nielsen J (2008) Metabolic footprinting in microbiology: methods and applications in functional genomics and biotechnology. Trends Biotechnol 26: 490–497.
9. von Roepenack-Lahaye E, Degenkolb T, Zerjeski M, Franz M, Roth U, et al. (2004) Profiling of Arabidopsis secondary metabolites by capillary liquid chromatography coupled to electrospray ionization quadrupole time-of-flight mass spectrometry. Plant Physiol 134: 548–559.
10. Saito K, Matsuda F (2008) Metabolomics for Functional Genomics, Systems Biology, and Biotechnology. Annu Rev Plant Biol 16.
11. Oksman-Caldentey KM, Saito K (2005) Integrating genomics and metabolomics for engineering plant metabolic pathways. Curr Opin Biotechnol 16: 174–179.
12. Saito K, Hirai MY, Yonekura-Sakakibara K (2008) Decoding genes with coexpression networks and metabolomics - 'majority report by precogs'. Trends Plant Sci 13: 36–43.
13. Powers R (2007) Functional genomics and NMR spectroscopy. Comb Chem High Throughput Screen 10: 676–697.
14. Wu L, van Winden WA, van Gulik WM, Heijnen JJ (2005) Application of metabolome data in functional genomics: a conceptual strategy. Metab Eng 7: 302–310.
15. Perera V, De Torres Zabala M, Florance H, Smirnoff N, Grant M, et al. (2012) Aligning extracted LC-MS peak lists via density maximization. Metabolomics 8: 175–185.
16. Smith C, Want EJ, O'Maille G, Abagyan R, Siuzdak G (2006) XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. Analytical chemistry 78: 779–787.
17. De Souza D, Saunders EC, McConville MJ, Likic VA (2006) Progressive peak clustering in GC-MS Metabolomic experiments applied to Leishmania parasites. Bioinformatics 22: 1391–1396.
18. Robinson M, De Souza DP, Keen WW, Saunders EC, McConville MJ, et al. (2007) A dynamic programming approach for the alignment of signal peaks in multiple gas chromatography-mass spectrometry experiments. BMC Bioinformatics 8: 419.
19. Lange E, Gropl C, Schulz-Trieglaff O, Leinenbach A, Huber C, et al. (2007) A geometric approach for the alignment of liquid chromatography-mass spectrometry data. Bioinformatics 23: i273–281.
20. Fischer B, Grossmann J, Roth V, Gruissem W, Baginsky S, et al. (2006) Semi-supervised LC/MS alignment for differential proteomics. Bioinformatics 22: e132–140.
21. de Groot J, Fiers MW, van Ham RC, America AH (2008) Post alignment clustering procedure for comparative quantitative proteomics LC-MS data. Proteomics 8: 32–36.
22. Chae M, Reis RJS, Thaden JJ (2008) An iterative block-shifting approach to retention time alignment that preserves the shape and area of gas chromatography-mass spectrometry peaks. BMC Bioinformatics 9: S15.
23. Johnson K, Wright BW, Jarman KH, Synovec RE (2003) High-speed peak matching algorithm for retention time alignment of gas chromatographic data for chemometric analysis. J Chromatogr A 996: 141–155.
24. Hoffmann N, Stoye J (2009) ChromA: signal-based retention time alignment for chromatography-mass spectrometry data. Bioinformatics 25: 2080–2081.
25. Katajamaa M, Oresic M (2005) Processing methods for differential analysis of LC/MS profile data. BMC Bioinformatics 6: 179.
26. Duran A, Yang J, Wang L, Sumner LW (2003) Metabolomics spectral formatting, alignment and conversion tools (MSFACTs). Bioinformatics 19: 2283–2293.
27. Tibshirani R, Hastie T, Narasimhan B, Soltys S, Shi G, et al. (2004) Sample classification from protein mass spectrometry, by 'peak probability contrasts'. Bioinformatics 20: 3034–3044.
28. Voss B, Hanselmann M, Renard BY, Lindner MS, Köthe U, et al. (2011) SIMA: simultaneous multiple alignment of LC/MS peak lists. Bioinformatics 27: 987–993.
29. Lommen A (2009) MetAlign: interface-driven, versatile metabolomics tool for hyphenated full-scan mass spectrometry data preprocessing. Anal Chem 81: 3079–3086.
30. Sturm M, Bertsch A, Gropl C, Hildebrandt A, Hussong R, et al. (2008) OpenMS - an open-source software framework for mass spectrometry. BMC Bioinformatics 9: 163.
31. Fiehn O, Wohlgemuth G, Scholz M (2005) Setup and Annotation of Metabolomic Experiments by Integrating Biological and Mass Spectrometric Metadata. Lecture Notes in Bioinformatics 3615: 224–239.
32. Baran R, Kochi H, Saito N, Suematsu M, Soga T, et al. (2006) MathDAMP: a package for differential analysis of metabolite profiles. BMC Bioinformatics 7: 530.
33. Schulz-Trieglaff O, Pfeifer N, Gropl C, Kohlbacher O, Reinert K (2008) LC-MSsim–a simulation software for liquid chromatography mass spectrometry data. BMC Bioinformatics 9: 423.
34. Wong J, Cagney G, Cartwright HM (2005) SpecAlign–processing and alignment of mass spectra datasets. Bioinformatics 21: 2088–2090.
35. Broeckling CD, Reddy IR, Duran AL, Zhao X, Sumner LW, et al. (2006) MET-IDEA: data extraction tool for mass spectrometry-based metabolomics. Anal Chem 78: 4334–4341.
36. Sedgewick C (1997) Algorithms in C, Parts 1–4. Boston, MA, USA: Addison-Wesley Professional.
37. Tusher V, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. PNAS 98: 5116–5121.
38. Efron B, Tibshirani R, Storey JD, Tusher V (2001) Empirical Bayes analysis of a microarray experiment. Journal of American Statistical Association 96: 1151–1160.
39. Baldi P, Long AD (2001) A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. Bioinformatics 17: 509–519.
40. Conrads T, Anderson GA, Veenstra TD, Pasa-Tolić L, Smith RD (2000) Utility of accurate mass tags for proteome-wide protein identification. Anal Chem 72: 3349–3354.
41. Norbeck A, Monroe ME, Adkins JN, Anderson KK, Daly DS, et al. (2005) The utility of accurate mass and LC elution time information in the analysis of complex proteomes. J Am Soc Mass Spectrom 16: 1239–1249.
42. Skiena S, editor (2008) The Algorithm Design Manual: Springer.
43. Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W. H. Freeman.